UCRL-PRES-209469

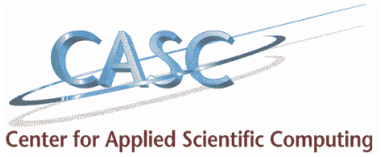
Parallel AMR Application Development with the SAMRAI Library

Andy Wissink & Rich Hornung

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
www.llnl.gov/CASC/SAMRAI

February 13, 2005
SIAM CSE Meeting

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.



SAMRAI Project Members

Noah Elliott

Brian Gunney

Rich Hornung

Photo unavailable

Craig Kapfer

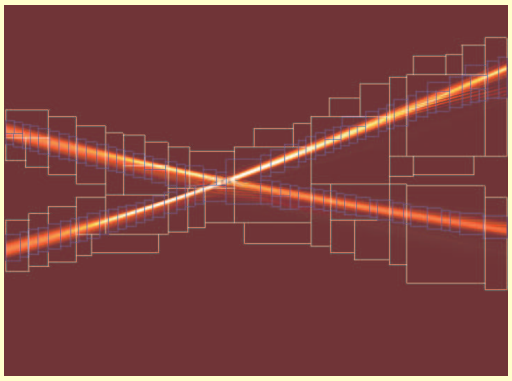
Steve Smith

Andy Wissink

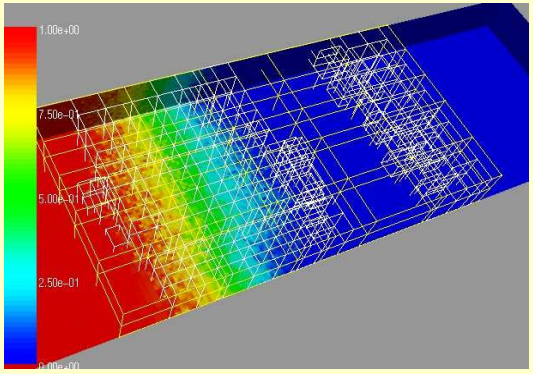
SAMRAI

Structured Adaptive Mesh Refinement Application Infrastructure

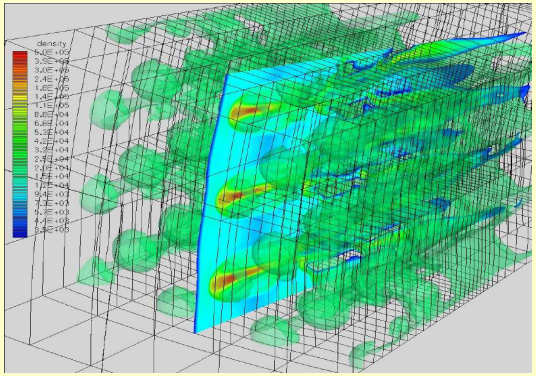
- AMR dynamically increases spatial and temporal grid resolution to resolve important local features
- SAMRAI is an object-oriented C++ framework that supports applications investigating multi-scale phenomena.
- Framework provides high-level reusable code and algorithms shared across a variety of applications.



Two beam laser interaction
ALPS



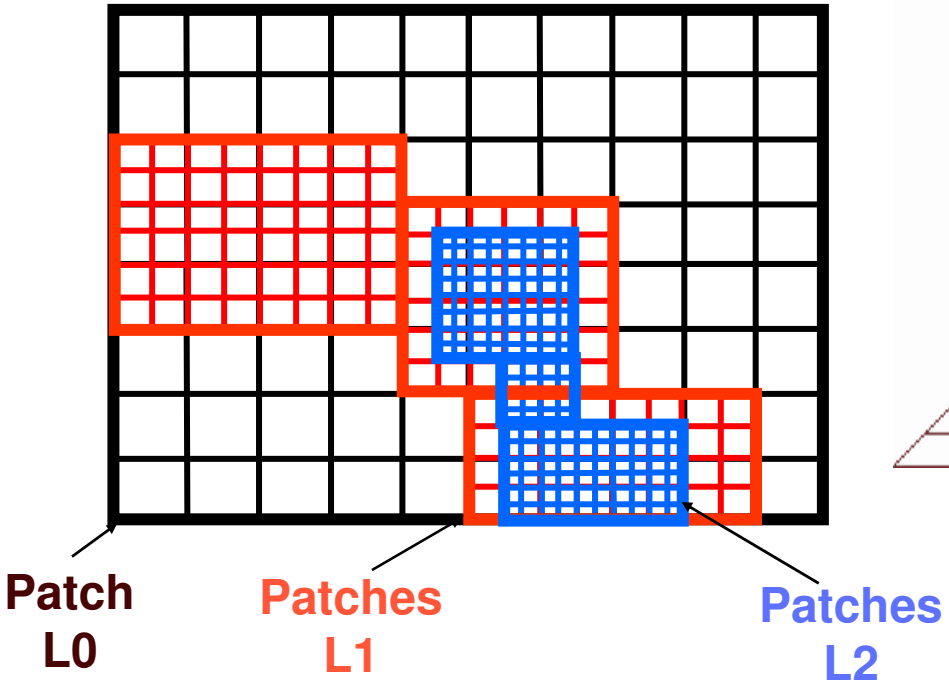
Richtmeyer-Meshkov instability
Hybrid Continuum Particle

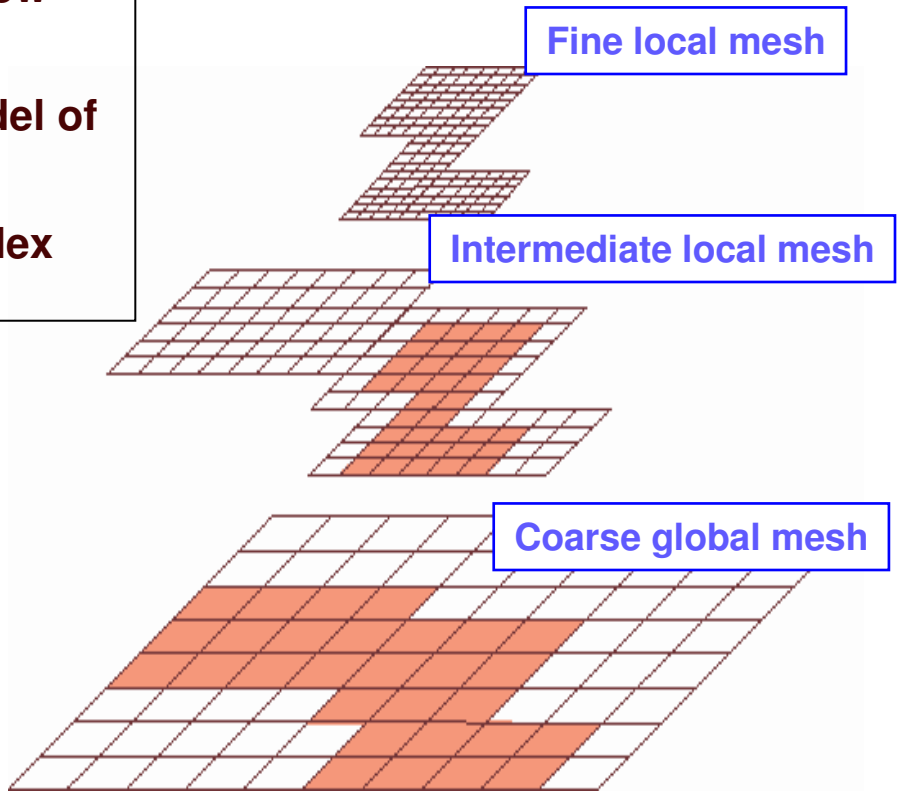


Richtmeyer-Meshkov instability
ALE-AMR

Structured AMR (SAMR) employs a "patch" hierarchy

- Hierarchy of nested "patch" levels → low overhead mesh description
- Data mapped to patches → simple model of data locality
- Patches cover logically rectangular index space



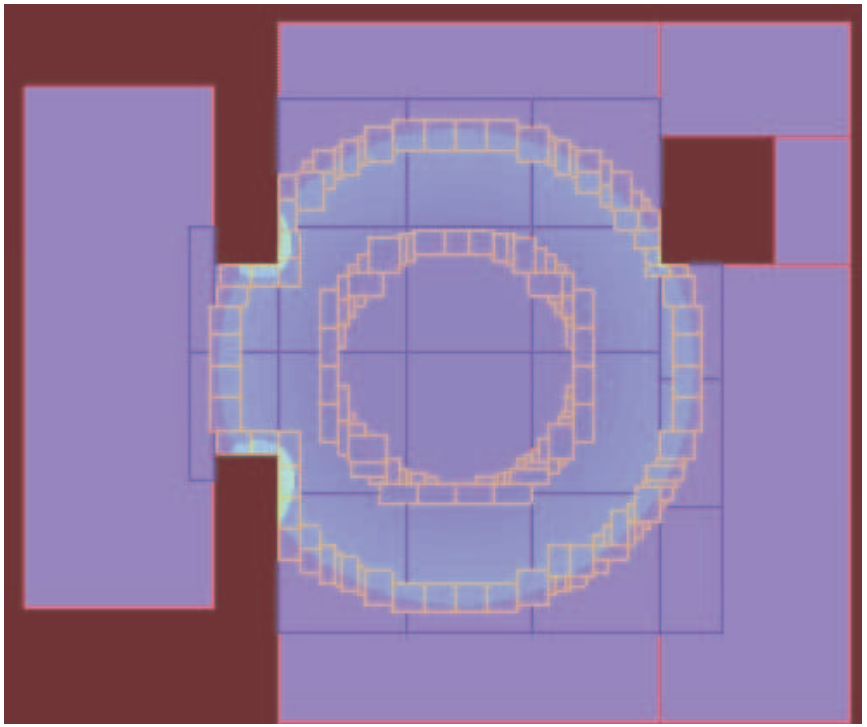


Berger, Oliger, Colella

SAMRAI manages many of the complexities of SAMR implementations

SAMRAI Provides:

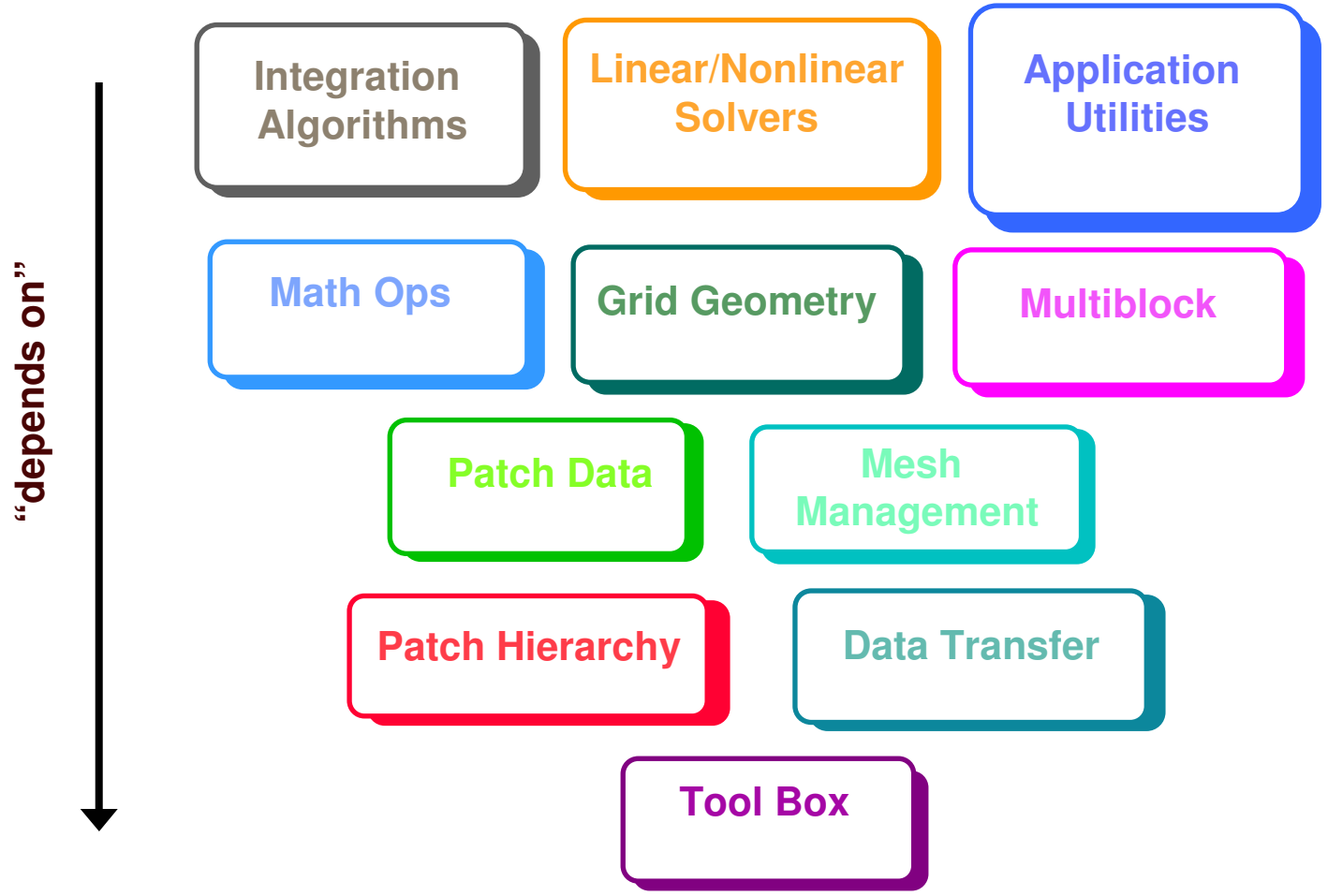
- Parallel communication (MPI)
- Dynamic gridding support
- Inter-patch data transfer operations (copy, coarsen, refine, time int, …)
- Solver interfaces for SAMR data (PetSc, hypre, pvode)
- Checkpointing and restart (HDF5)
- Visualization support (VisIt)



User provides:

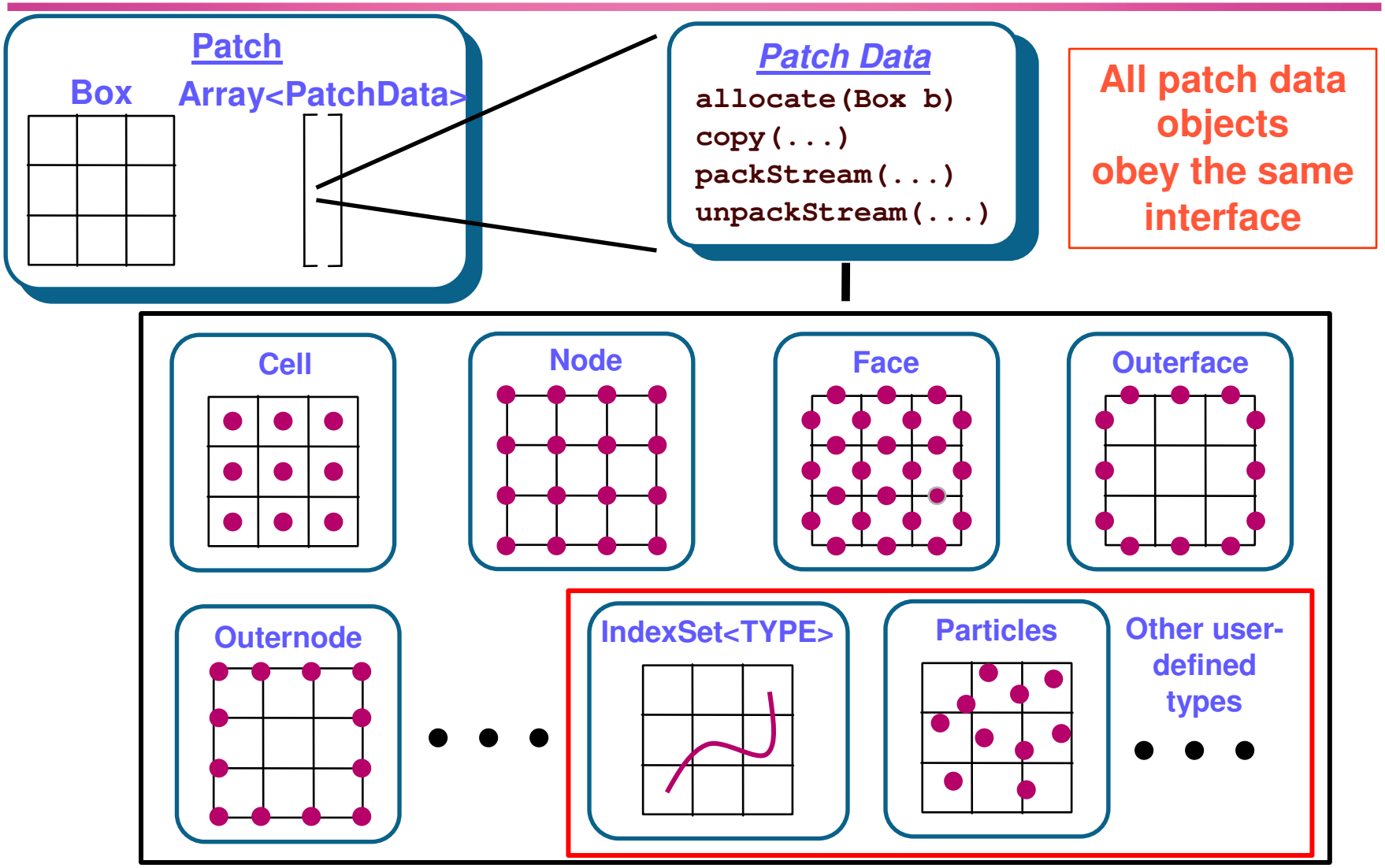
- (serial) numerical routines for individual patches
- Composition of SAMRAI classes to implement desired algorithm.

SAMRAI is an object-oriented "toolbox" of classes for SAMR application development



Hornung, Kohn, "Managing Application Complexity in the SAMRAI Object-oriented Framework", *Concurrency Computat.: Pract. Exper.* **14**:347-368 (2002)

A SAMRAI "patch" contains all data on a box region of the computational mesh



SAMRAI *Variable* and *PatchData* delineate "static" and "dynamic" data concepts

Solution algorithms and variables tend to be static

Variable object

- defines a data quantity; type, (centering), (depth)
- attributes:
  - name (string)
  - unique instance id (int)
- *Variable* objects generally persist throughout computation

Mesh and data objects tend to be dynamic

PatchData object

- represents data on a "box"
- attributes:
  - box
  - ghost cell width
- Attributes facilitate construction of communication dependencies
- *PatchData* objects are created and destroyed as mesh changes

Comm. Algorithm and *Schedule:* "static" and "dynamic" communication concepts

Solution algorithms and variables tend to be static

- **Communication Algorithm**
  - describes data transfer phase of computation
  - expressed using variables, operators, …
  - independent of mesh
  - typically persists throughout computation

Mesh and data objects tend to be dynamic

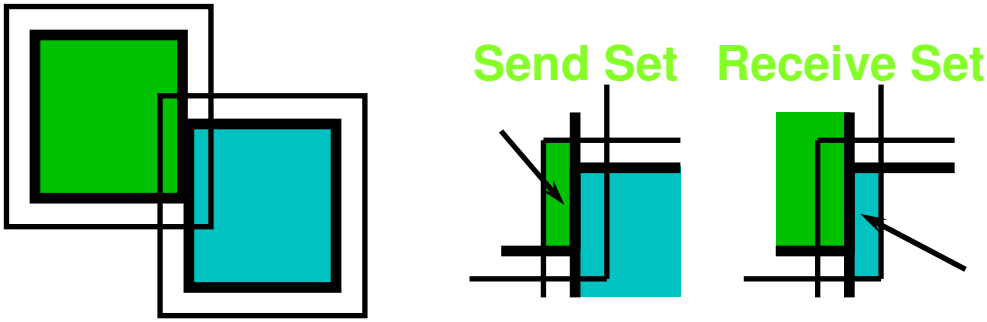
- **Communication Schedule**
  - manages details of data movement on mesh
  - created by communication algorithm
  - depends on mesh
  - re-created when mesh changes

Compare with...

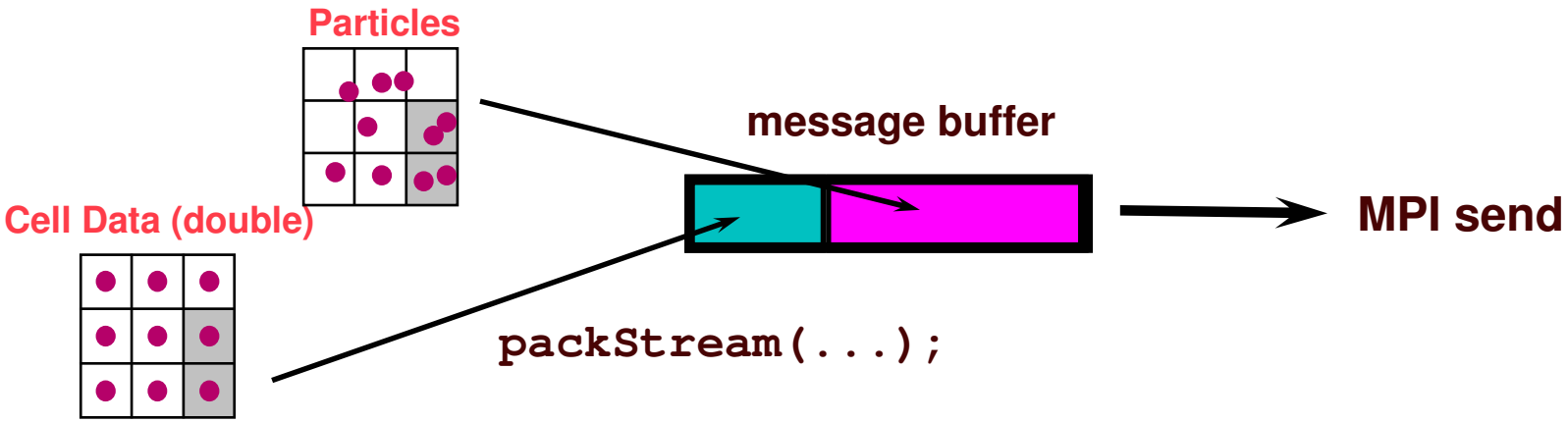
- ***Variable***
  - defines a data quantity independent of mesh
  - usually persists throughout computation
- **PatchData**
  - represents data on a "box"
  - created and destroyed as mesh changes

Communication schedules create and store data dependencies

- **Amortize cost of creating send/receive sets over multiple communication cycles**

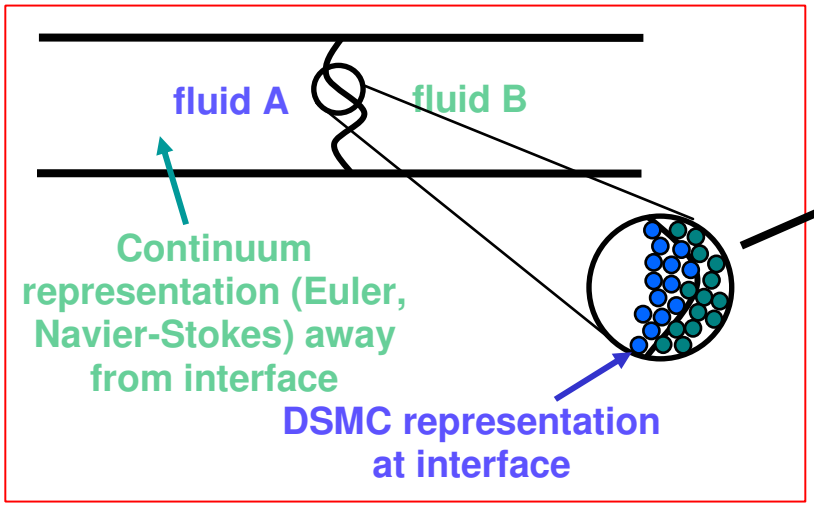


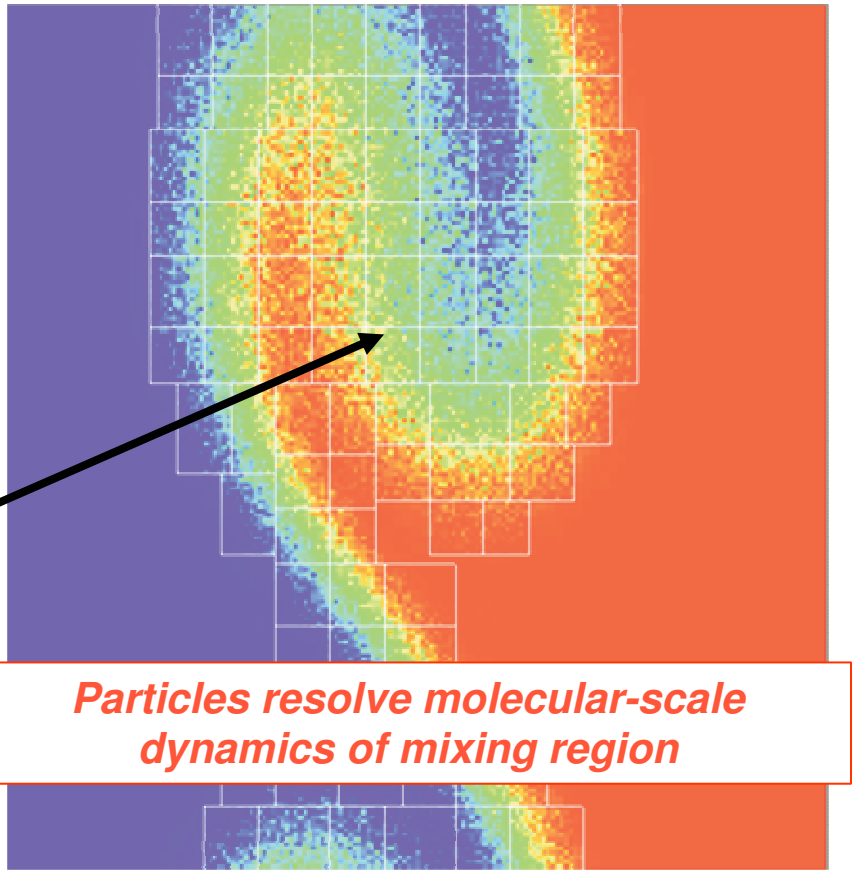
- **Data from various sources packed into single message stream**
  - supports complicated variable-length data
  - one send per processor pair (low latency)



Adaptive Mesh and Algorithm Refinement (AMAR) refines mesh and numerical model

- AMR is used to refine continuum calculation and focus particles
- Algorithm switches to discrete atomistic method to include physics absent in continuum model

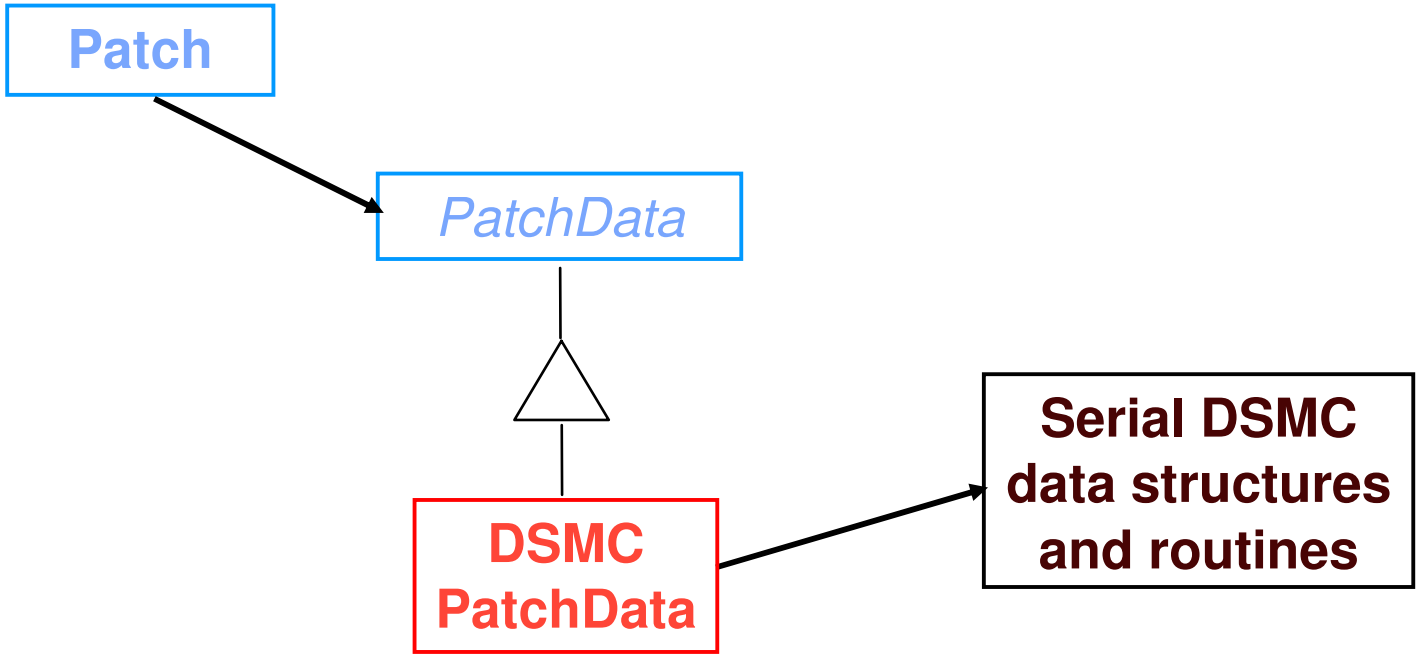




Particles resolve molecular-scale dynamics of mixing region

Wijesinghe, Hornung, Garcia, Hadjiconstantinou, ""Three-dimensional Hybrid Continuum-Atomistic Simulations for Multiscale Hydrodynamics", *J. Fluid. Eng.*, ***126***:768-777 (Sept 2004).

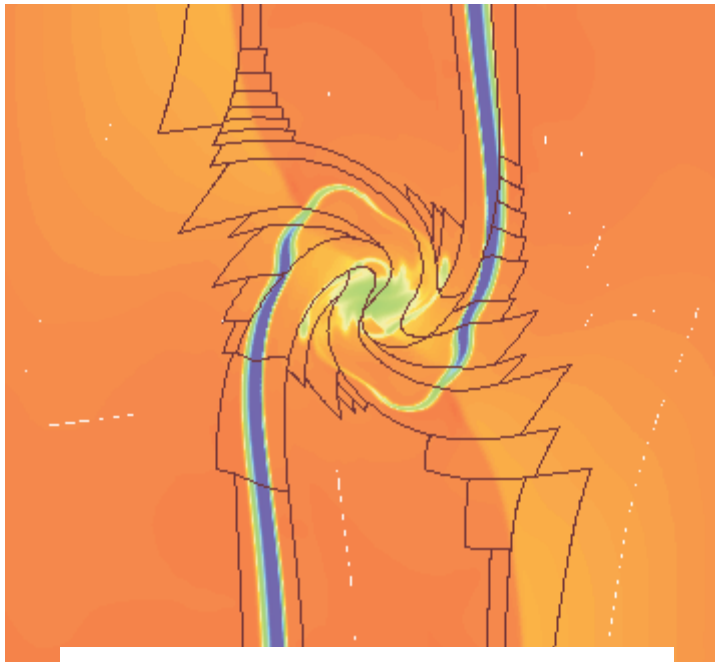
Pre-existing particle data structures coupled to SAMRAI via patch data interface



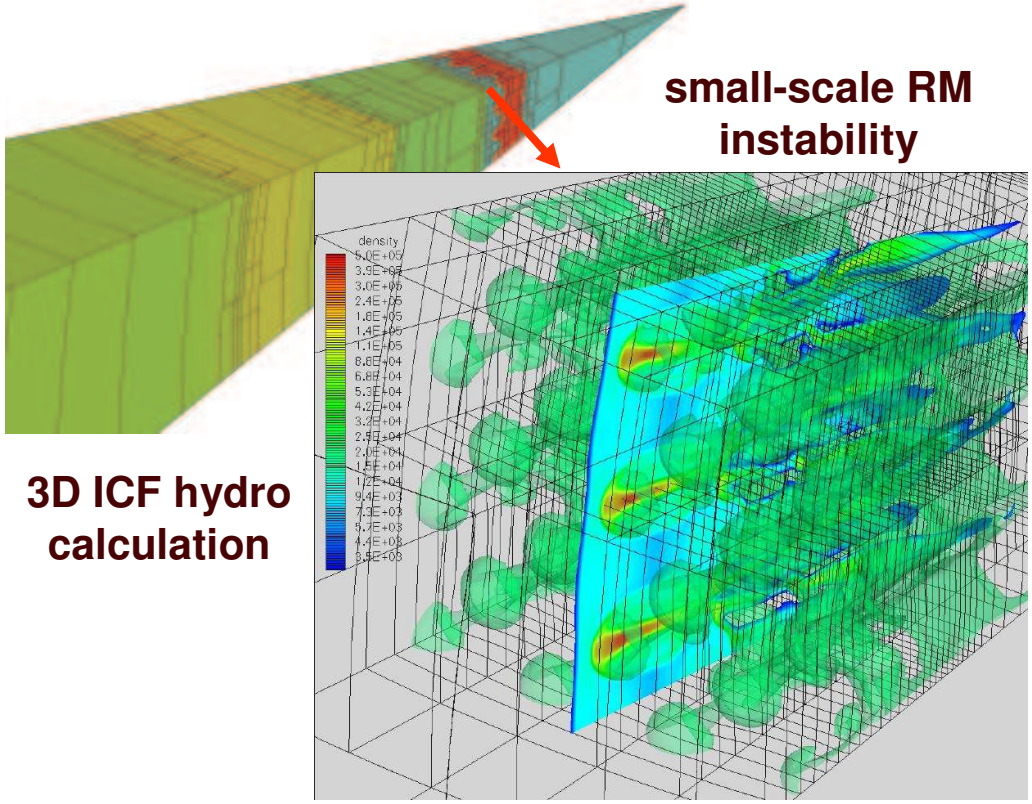

```
DsmcPatchData* particles = patch->getPatchData(. . .);
particles->advance(dt);
```

ALE-AMR combines ALE integration with AMR

- Advantages of ALE (multiple materials, moving interfaces)
- Advantages of AMR (dynamic addition & removal of mesh points)



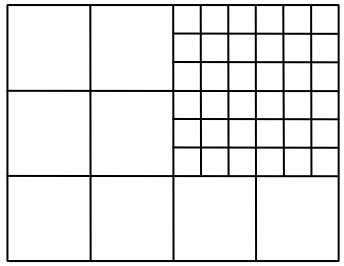
Moving-deforming AMR grid



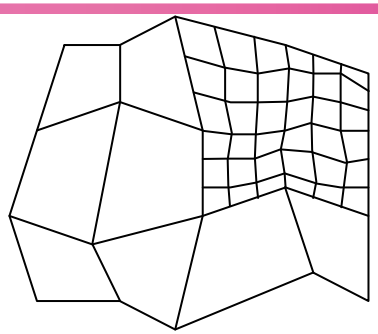
3D ICF hydro calculation

Anderson, Elliott, Pember, "An Arbitrary Lagrangian-Eulerian Methods with Adaptive Mesh Refinement for the Solution of the Euler Equations", *J. Comp. Phys.* **199**(2): 598-617 (2004).

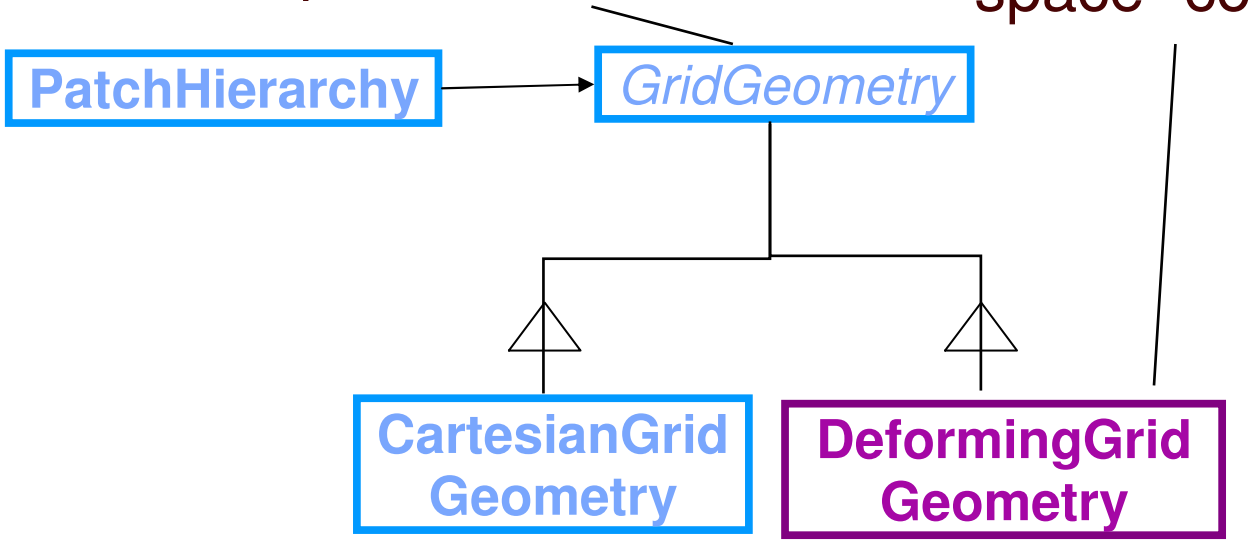
Deforming grids in ALE-AMR managed by specializing SAMRAI grid geometry



Manages "index space" coordinates

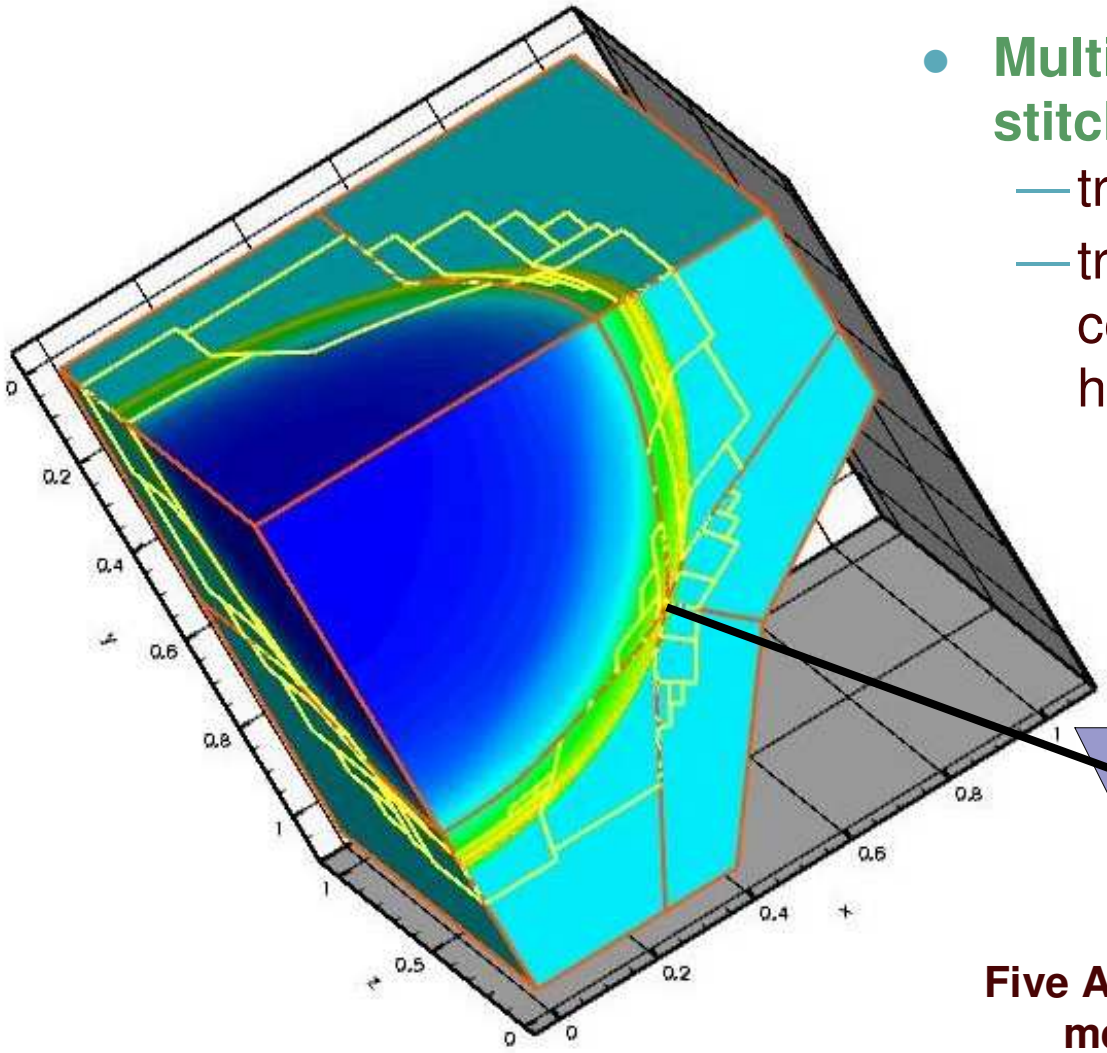


Manages "physical space" coordinates

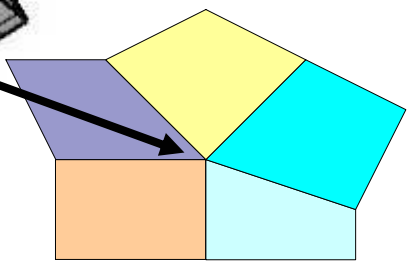


Utilizes all the other features of SAMRAI:
- parallel communication
- adaptive gridding
- solver interfaces
- etc.

SAMRAI "multiblock" capability supports multiple patch hierarchies



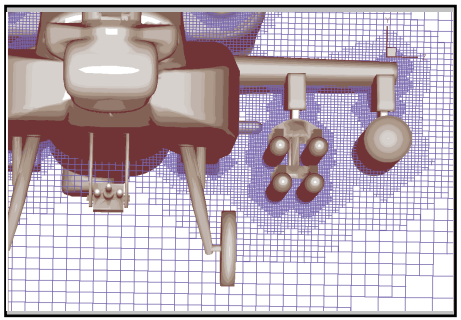
- **Multiple index spaces stitched together**
  - translations & rotations
  - transparent data communication between hierarchies



Five AMR patch hierarchies meet at single point

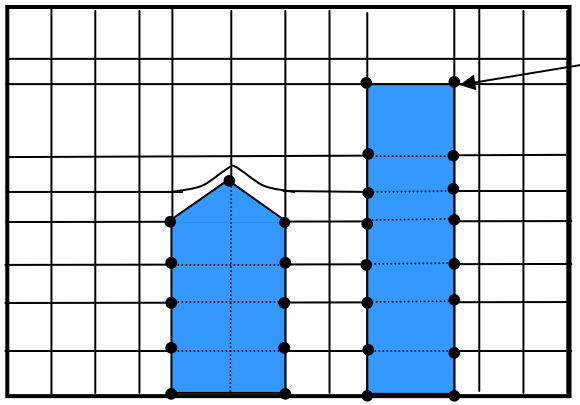
SAMRAI supports Cartesian Embedded Boundary grid representations

- **Constructing body-fitted logically rectangular grids is tedious and expensive.**
- **Embedded boundary grids constructed automatically in SAMRAI**
  - **Built from polygons or from surface triangulation using CUBES**

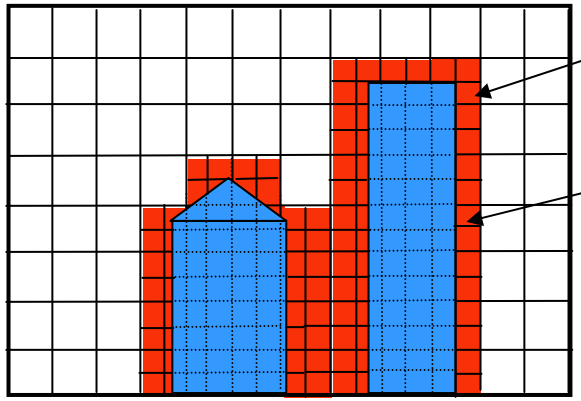


Example embedded boundary mesh constructed with CUBES

M. Berger, Courant Inst./NASA Ames



Body-fitted grid



Embedded Boundary Grid

SAMRAI index data supports embedded boundary as "patch data"

- ***IndexVariable*** **and** ***IndexData*** **classes manage data quantities on irregular index sets**

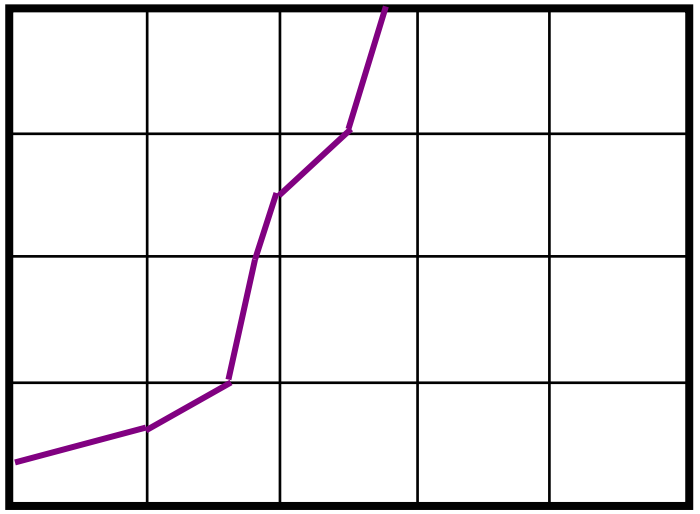
```
IndexVariable<TYPE> ivar("name")
IndexData<TYPE> idata(Box& box, ghosts)
```

"TYPE"

Required methods

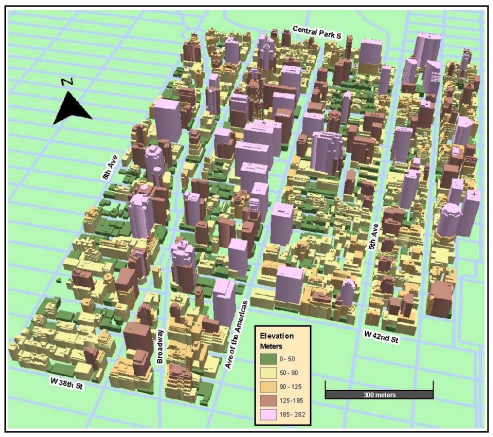
```
TYPE()
TYPE& operator=(const TYPE&)
getDataStreamSize(Box&)
packStream(...)
unpackStream(...)
```

e.g.



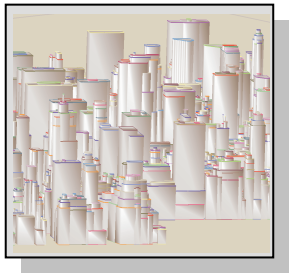
CutCell type describes internal boundary and state information along boundary

AUDIM applying adaptive meshing for CFD Urban Dispersion Modeling



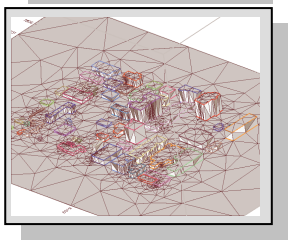
"Shapefile" (Manhattan)

Eleven (Overture)
Polygonal Geometry Representation



polygons

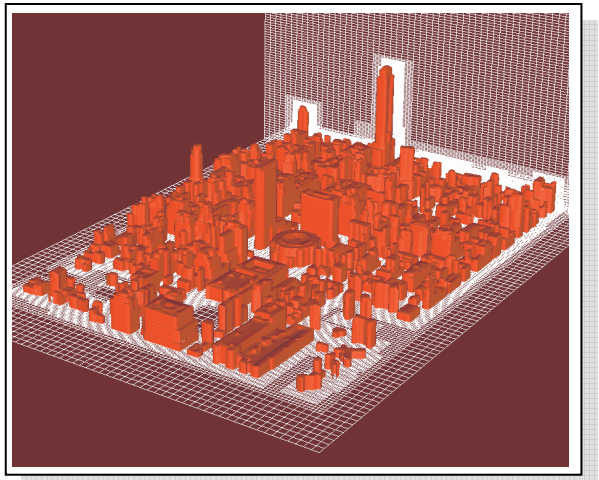
CUBES (Berger)



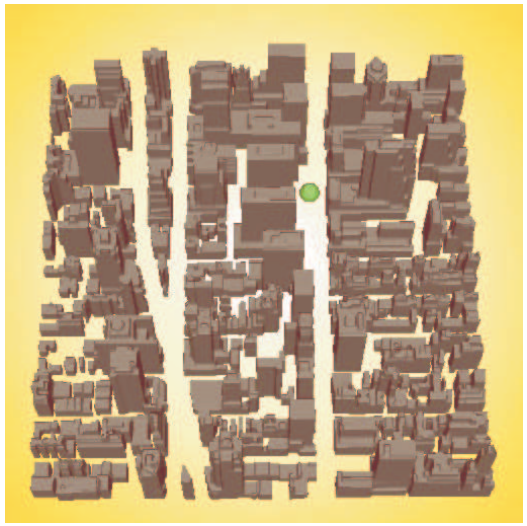
Triangulated surf grid

SAMRAI
Adaptive Meshing, embedded boundary representation, parallel AMR support

FEM3MP
FE based CFD Dispersion model

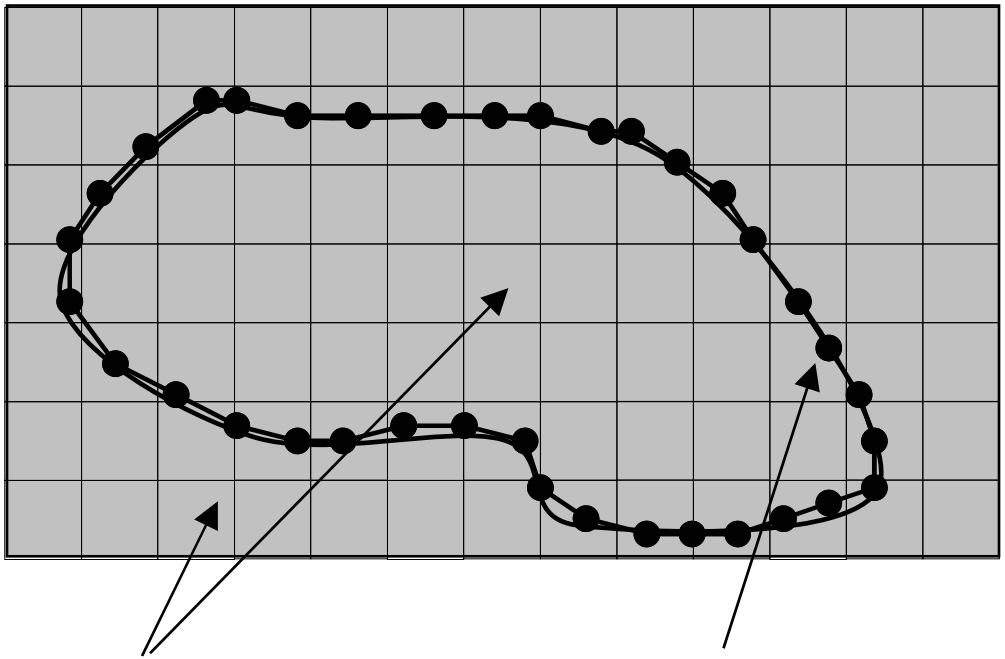


AMR Emb Boundary Vol Mesh



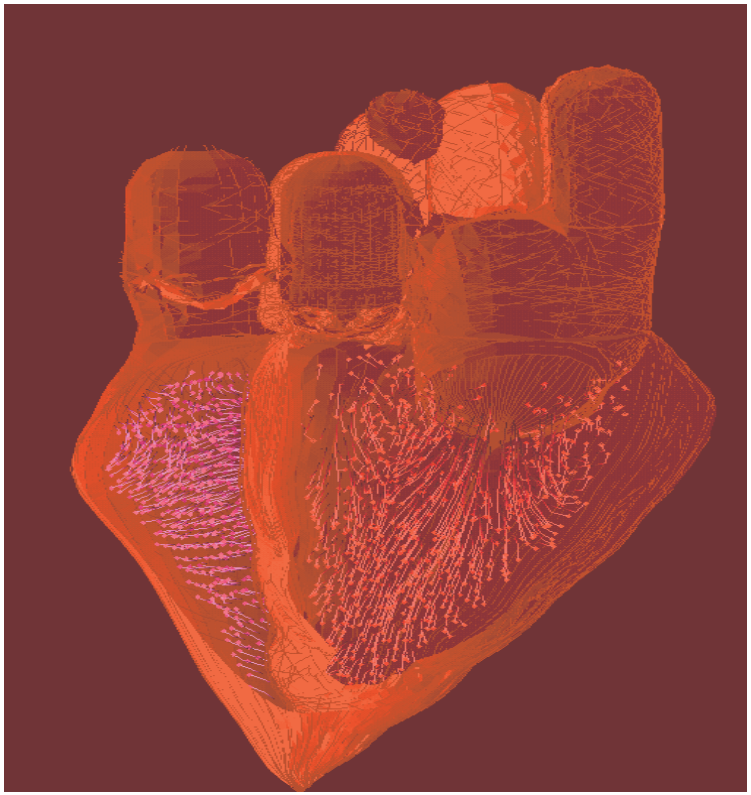
Adaptive CFD simulation

Immersed boundary methods model fluid structure interactions



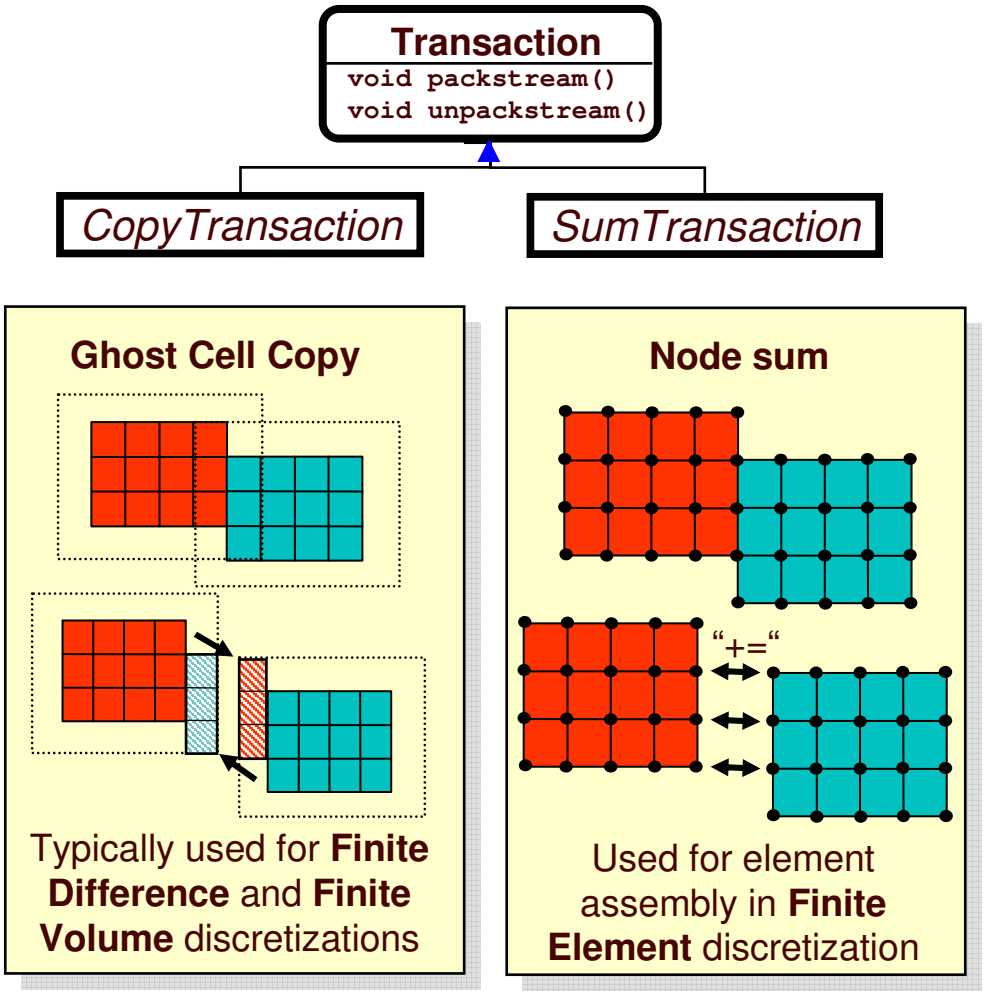
Fluid domain:
$\mathbf{u}(\mathbf{x},t),p(\mathbf{x},t),\mathbf{f}(\mathbf{x},t)$

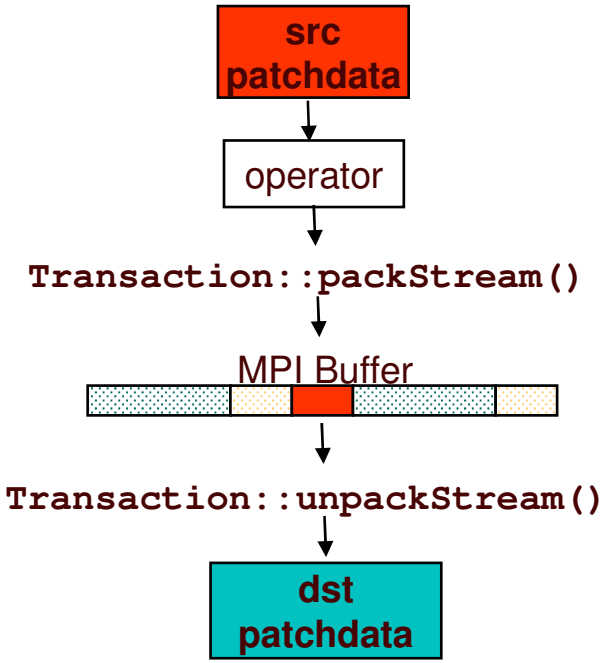
Structure domain:
$\mathbf{X}(s,t)$, $\mathbf{F}(s,t)$



Griffith, Peskin (NYU) are developing an electrical-mechanical heart model combining immersed boundaries and AMR (SAMRAI)

Communication specialized for finite-element based operations

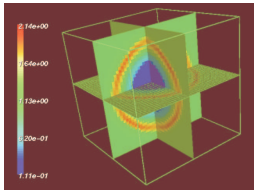




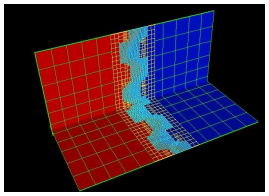
- **"Sum" transactions used for finite element calculations**
  - Used across multiple levels for AMR
  - Node & Edge sum available

SAMRAI supports applications on large parallel platforms

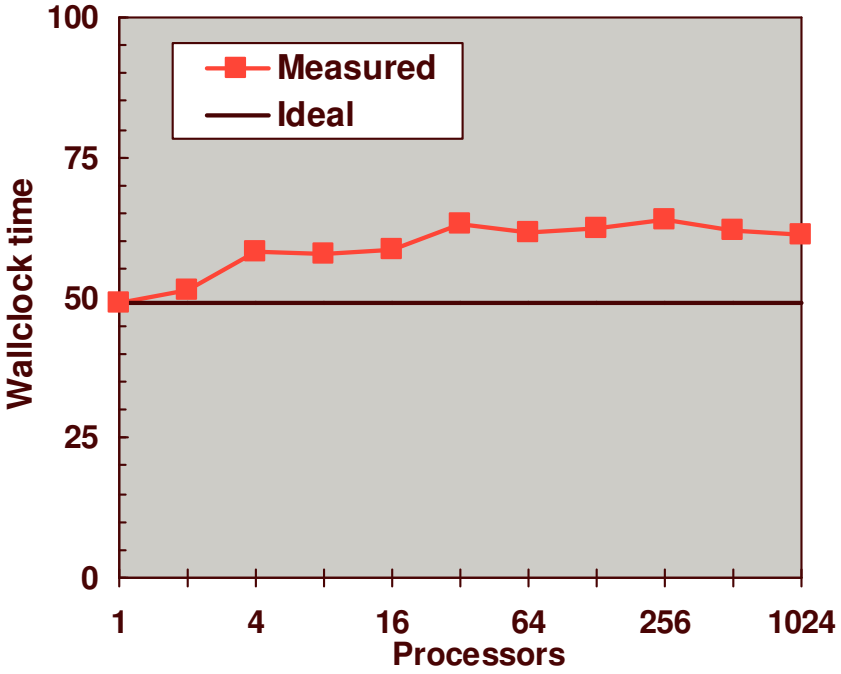
Scaled Euler Hydrodynamics
IBM Blue Pacific



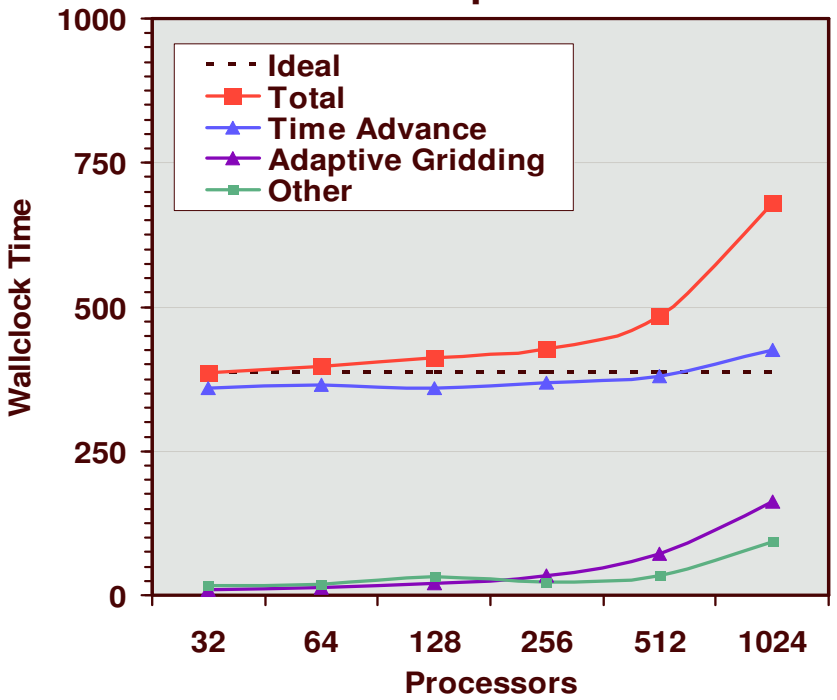
Scaled Linear Advection
Linux MCR Cluster



Non-adaptive



Adaptive



Wissink, Hysom, Hornung, "Enhancing Scalability of Parallel Structured AMR Calculations", *2003 Int. Conf. on Supercomputing (ICS03), San Francisco, CA,* June 2003, pp. 336-347.

Concluding remarks

- **AMR is an important technology for large-scale science & engineering problems that require greater resolution of localized features**
- **New applications require expansion of current AMR methodologies**
  - Model refinement in addition to grid refinement
  - Support for variety of data representations and non-Cartesian grids
  - Complex geometries
  - Efficiency on large-scale parallel architectures
- **AMR libraries must effectively interoperate with other software packages – solver libraries, grid generation packages, etc.**